

---

---

# Kwanty

Jacek Markiewicz

2025/2026

---

---

Imagination will often carry us to worlds that never were. But without it we go nowhere.

*Carl Sagan*

## Spis treści

<b>1</b>	<b>Algorytm Deutsch-Jozsa</b>	<b>2</b>
1.1	Bernstein-Vazirani . . . . .	3
<b>2</b>	<b>Algorytm Grovera</b>	<b>5</b>
2.1	Amplitude Amplification . . . . .	6
<b>3</b>	<b>Kubityzacja</b>	<b>7</b>
3.1	LCU . . . . .	7
<b>4</b>	<b>QFT</b>	<b>10</b>
4.1	Quantum Phase Estimation . . . . .	11
4.2	Phase kickback . . . . .	12
4.3	Hamiltonian simulations . . . . .	12
<b>5</b>	<b>Algorytm Shor'a</b>	<b>14</b>
5.1	Algorytm Shora . . . . .	15
5.2	Test Hadamarda . . . . .	15
5.3	Test SWAP . . . . .	17
5.4	Amplitude estimation . . . . .	18
<b>6</b>	<b>Stabilizatory</b>	<b>19</b>
<b>7</b>	<b>Odporność na błędy</b>	<b>20</b>
<b>8</b>	<b>Pamięć kwantowa</b>	<b>21</b>

# 1 Algorytm Deutsch-Jozsy

## Problem

Dana jest czarna skrzynka w postaci funkcji  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , o której wiemy, że zachodzi jedno z poniższych:

- $f$  jest funkcją *stałą* (zawsze zwraca 0 lub zawsze zwraca 1) lub
- $f$  jest funkcją *zbalansowaną* (przypisuje 0 takiej samej ilości elementów co wartość 1)

Naszym zadaniem jest stwierdzenie, czy  $f$  jest stała czy zbalansowana.

Chcemy dodatkowo dokonać tego wykonując możliwie mało ewaluacji funkcji  $f$ .

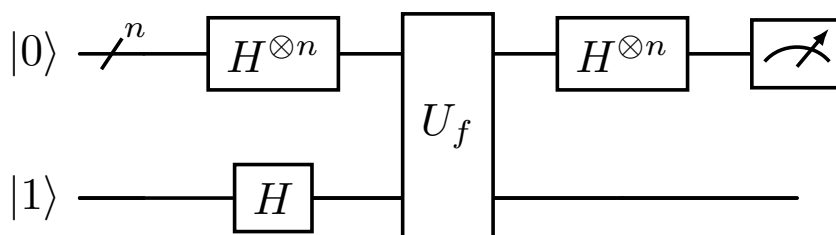
## Klasyczne podejście

Na komputerze klasycznym konieczne jest  $\mathcal{O}(2^n)$  operacji. Wynika to z tego, że pesymistycznie pierwsze  $2^{n-1}$  sprawdzeń wartości może zakończyć się tym samym wynikiem, niezależnie jakiego rodzaju jest to funkcja.

Innym pomysłem jest algorytm randomizowany, tj. wylosowanie  $k$  wartości i ich zewaluowanie. Jeśli otrzymaliśmy zarówno 0 jak i 1 jako wyniki, to funkcja na pewno była zbalansowana. Wpp. strzelamy, że była funkcją stałą. Jest to niezła metoda, gdyż prawdopodobieństwo pomyłki (czyli uznania  $f$  za stałą gdy była tylko zbalansowaną) wynosi  $2^{-k}$ . Wymagało to dodatkowo tylko  $k$  ewaluacji funkcji  $f$ .

## Kwantowe podejście

My jednak będziemy jeszcze lepsi i stworzymy algorytm deteministyczny, który da nam zawsze poprawną odpowiedź wykonując tylko jedną ewaluację funkcji  $f$ . Bez owijania w bawełnę oto nasz szukany obwód kwantowy:

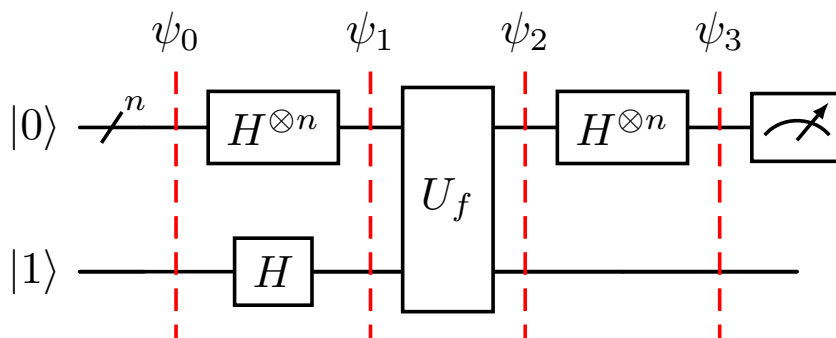


(1)

Jeśli zmierzmy  $|0\rangle^{\otimes n}$  to funkcja była stała. W przeciwnym przypadku była zbalansowana.

### Dowód.

Żeby zobaczyć dlaczego ten układ rozwiązuje nasz problem musimy przeanalizować co się dzieje w każdym jego kroku:



(2)

- $|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$
- $|\psi_1\rangle$ :

Przypomnijmy sobie jak działa bramka Hadamarda:

$$H |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$

$$H |1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

$$|\psi_1\rangle = H|\psi_0\rangle = (H|0\rangle^{\otimes n})(H|1\rangle) = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0\rangle - |1\rangle)$$

- $|\psi_2\rangle$ :

Wyroczenia  $U_f$  zachowuje się tak:

$$U_f |x\rangle_n |y\rangle = |x\rangle_n |y \oplus f(x)\rangle$$

$$\Rightarrow |\psi_2\rangle = U_f |\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|f(x)\rangle - |1 \oplus f(x)\rangle)$$

ale  $f(x)$  przyjmuje tylko wartości 0 lub 1, zatem możemy zapisać to sprytniej

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle)$$

- $|\psi_3\rangle$ :

Bramka Hadamarda potrafi jeszcze tak:

$$H|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{x \cdot i}$$

gdzie  $x \cdot i$  oznacza iloczyn skalarny, czyli  $x \cdot i = x_1 i_1 \oplus \dots \oplus x_n i_n$ .

Ostatni wynikowy kubit nie jest nam potrzebny, więc go ignorujemy. Zatem niech  $\psi'_2$  oznacza stan  $\psi_2$  z uciętym ostatnim bitem, analogicznie  $\phi'_3$ . Wtedy:

$$|\psi'_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$$

Zatem ostatnia transformacja daje nam

$$\begin{aligned} |\psi'_3\rangle &= H|\psi'_2\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \cdot \sum_{i=0}^{2^n-1} (-1)^{x \cdot i} |i\rangle \\ &= \frac{1}{2^n} \sum_{i=0}^{2^n-1} \left[ \sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot i} \right] \end{aligned}$$

Ale teraz zauważmy, że prawdopodobieństwo zmierzenia stanu  $0^{\otimes n}$  wynosi

$$\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2$$

Wystarczy teraz przeanalizować przypadki:

1. Jeśli  $f$  jest stała, to  $\sum_{x=0}^{2^n-1} (-1)^{f(x)} = \pm 2^n$ , czyli prawdopodobieństwo wyjdzie równe 1.
2. Jeśli  $f$  jest zbalansowana, to  $\sum_{x=0}^{2^n-1} (-1)^{f(x)} = 0$ , bo wyrazów sumy równych 1 jest tyle samo co równych -1. Zatem prawdopodobieństwo wyniesie 0.

Czyli by stwierdzić czy  $f$  jest stała wystarczy sprawdzić czy na końcu zmierzmy  $0^{\otimes n}$ . Jeśli zmierzmy cokolwiek innego to znaczy, że była zbalansowana.

□

## 1.1 Bernstein–Vazirani

### Problem

Dana jest czarna skrzynka w postaci funkcji  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ . Mamy zapewnione, że  $f$  jest postaci

$$f(x) = x \cdot s$$

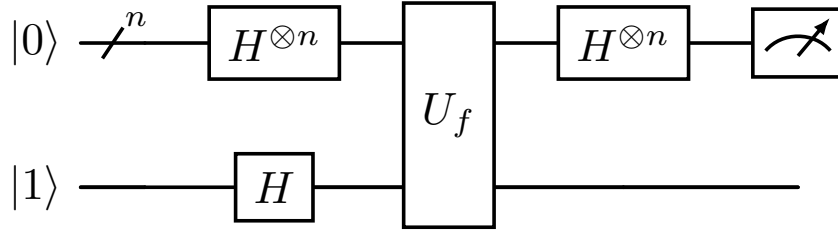
gdzie  $\cdot$  jest iloczynem skalarnym *mod* 2, a  $s$  - nieznanym wektorem z  $\{0, 1\}^n$ .

Zadanie polega na wyznaczeniu  $s$ .

Dodatkowo chcemy tego dokonać wykonując możliwie mało zapytań o funkcję  $f$ .

Łatwo pokazać, że klasyczne podejście zawsze wymaga co najmniej  $n$  zapytań o wartość  $f(x)$ . Wynika to z faktu, że przestrzeń potencjalnych wartości  $s$  jest wielkości  $2^n$ , a każde zapytanie się o wartość  $f(x)$  odrzuca pewną część tej przestrzeni, pesymistycznie co najwyżej połowę. Czyli aby być pewnym wartości  $s$  trzeba wykonać co najmniej  $\lg(2^n) = n$  zapytań.

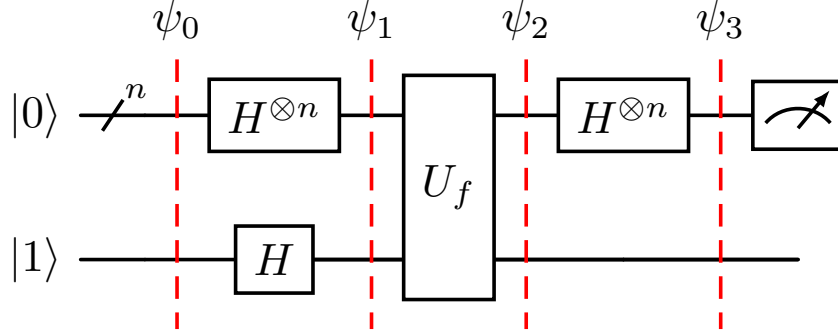
My będziemy lepsi i wyznaczmy deterministycznie wartość  $s$  używając tylko jednej ewaluacji funkcji  $f$ . By to osiągnąć użyjemy następującego układu:



(3)

Po jego uruchomieniu szukany przez nas wektor  $s$  magicznie pojawi się na mierzonym wyjściu.

**Dowód.** By to pokazać, musimy przeanalizować działanie układu.



(4)

A zatem po kolei:

- $|\psi_0\rangle = |0\rangle_n |1\rangle$
- Aplikujemy Hadamarda. Pamiętamy, że:

$$H|0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle, \quad H|1\rangle = |+\rangle$$

Zatem

$$|\psi_1\rangle = H|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |+\rangle$$

- Aplikujemy wyrocznie  $U_f$ . Wiemy, że działa ona tak:

$$U_f |x\rangle_n |y\rangle = |x\rangle_n |f(x)\rangle$$

Czyli:

$$U_f |x\rangle_n |+\rangle = U_f |x\rangle_n \left( \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = |x\rangle_n \left( \frac{|f(x)\rangle - |f(x) \oplus 1\rangle}{\sqrt{2}} \right)$$

Zauważmy, po "zignorowaniu" ostatniego kubitu jedynym widocznym wynikiem  $f(x)$  w układzie jest zmieniona faza (+1 lub -1).

Niech  $|g\rangle$  oznacza stan ostatniego bitu. Wtedy mamy:

$$|\psi_2\rangle = U_f |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle |g\rangle$$

- Ponownie Hadamard. Pamiętając, że  $H|y\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{k \cdot i} |i\rangle$  otrzymujemy:

$$|\psi_3\rangle = H|\psi_2\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{f(x)+x \cdot y} |y\rangle |g\rangle$$

Dla uproszczenia analizy zapomnijmy o kubicie  $|g\rangle$ . Zauważmy, że dla konkretnego  $y$ :

$$\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)+x \cdot y} = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x \cdot s+x \cdot y} = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{x \cdot (s \oplus y)}$$

Łatwo zauważyć, że gdy  $s = y$  to  $s \oplus y = 0$  i wyrażenie jest równe 1. Jeśli natomiast  $s \neq y$  to  $s \oplus y$  będzie przyjmowało wartości 0 i 1 dla tej samej ilości wyrazów, więc wyrażenie się wyzeruje.

Czyli prawdopodobieństwo zmierzenia stanu  $|s\rangle$  na pierwszych  $n$  kubitach wynosi 1.  $\square$

## 2 Algorytm Grovera

### Problem

Dana czarna skrzynka w postaci funkcji  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ . Funkcja ta zawsze przyjmuje wartość 0, poza jednym nieznanym argumentem  $\omega$ , dla którego  $f(\omega) = 1$ .

Zadanie polega na znalezieniu wartości  $\omega$ .

W przypadku klasycznym widać, że potrzebne jest średnio  $\frac{N}{2}$ , a pesymistycznie  $N - 1$  ewaluacji funkcji  $f$  do wyznaczenia  $\omega$ .

My dokonamy czegoś niesamowitego i wyznaczymy nieznaną wartość używając jedynie  $\mathcal{O}(\sqrt{N})$  ewaluacji  $f$ .

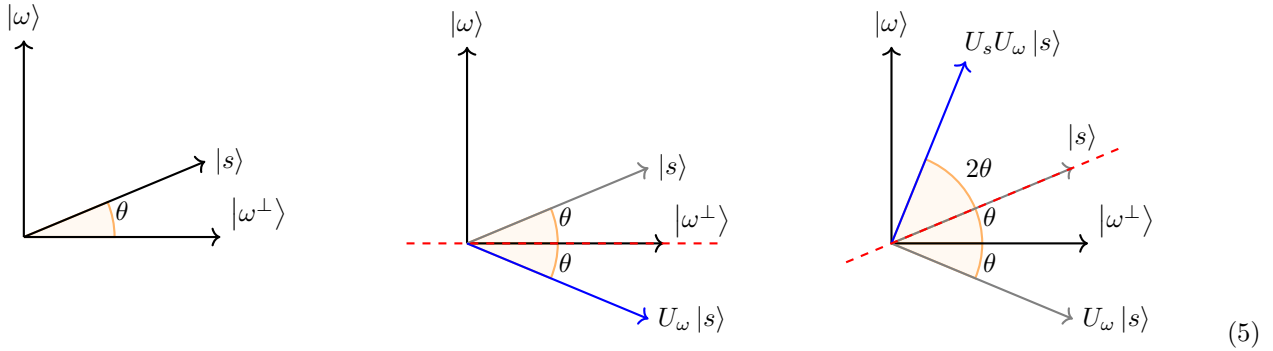
Użyjemy do tego dwóch operatorów:

- $U_\omega := I - 2|\omega\rangle\langle\omega|$
- $U_s := 2|s\rangle\langle s| - I$

Oba są odbiciami Householdera. Oznacza to tyle, że jeśli rozważymy płaszczyznę rozpiętą przez wektory  $|\omega\rangle$  i  $|\omega^\perp\rangle$  to:

- $U_\omega$  odbija każdy wektor względem płaszczyzny  $|\omega^\perp\rangle$  (prostopadłej do  $|\omega\rangle$ ).
- $U_s$  odbija wektor względem  $|s\rangle$

Załóżmy, że początkowo dany jest wektor  $|s\rangle$ , którego początkowy kąt z  $|\omega^\perp\rangle$  wynosi  $\theta$ . Wtedy przekształcenie  $U_s U_\omega$  graficznie wygląda następująco:



Przesunęliśmy się zatem w stronę  $|\omega\rangle$  o kąt  $2\theta$ . Okazuje się, że każde kolejne zaaplikowanie  $U_s U_\omega$  również przesuwamy nas w stronę  $|\omega\rangle$  o  $2\theta$ . Zatem możemy do skutku aplikować  $U_s U_\omega$ , a potem zmierzyć końcowy stan, który prawie na pewno będzie naszym szukanym  $|\omega\rangle$ .

Pozostaje jeszcze wybrać taki początkowy wektor  $|s\rangle$ , by  $\theta > 0$ . Z braku lepszych opcji możemy ustalić  $|s\rangle = H|0\rangle_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$ . Sprawdźmy ile wtedy wynosi  $\theta$ :

$$\cos \theta = \frac{\langle s | \omega^\perp \rangle}{\|s\| \cdot \|\omega^\perp\|} = \langle s | \omega^\perp \rangle$$

Wiemy też, że

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle = \frac{1}{\sqrt{N}} |\omega\rangle + \frac{1}{\sqrt{N}} \sum_{k=0, k \neq \omega}^{N-1} |k\rangle = \frac{1}{\sqrt{N}} |\omega\rangle + \frac{\sqrt{N-1}}{\sqrt{N}} |\omega^\perp\rangle$$

Zatem

$$\begin{aligned} \langle s | \omega^\perp \rangle &= \frac{1}{\sqrt{N}} \langle \omega | \omega^\perp \rangle + \frac{\sqrt{N-1}}{\sqrt{N}} \langle \omega^\perp | \omega^\perp \rangle = \frac{\sqrt{N-1}}{\sqrt{N}} \\ \Rightarrow \theta &= \arccos \sqrt{\frac{N-1}{N}} \end{aligned}$$

Z jedynki trygonometrycznej mamy

$$\theta = \arcsin \frac{1}{\sqrt{N}}$$

Ale dla małych kątów  $\arcsin x \approx x$ , czyli  $\theta \approx \frac{1}{\sqrt{N}}$ .

Za każdym razem przesuwamy się o kąt  $2\theta$ , a chcemy aby kąt z początkowego  $\approx 0$  stał się równy  $\pi/2$ . Łatwo zatem wyliczyć ilość potrzebnych zaaplikowań  $U_s U_\omega$ :

$$\frac{\pi/2}{2\theta} \approx \frac{\pi/2 \cdot \sqrt{N}}{2} = \boxed{\frac{\pi}{4} \sqrt{N}} = \mathcal{O}(\sqrt{N})$$

Mamy już zatem gotowy algorytm (Grovera):

1. Zainicjuj  $|s\rangle := H|0\rangle_n$
2. Zaaplikuj  $\frac{\pi}{4} \sqrt{N}$  razy  $U_s U_\omega$
3. Zmierz końcowy rejestr

### Definicja. (BPP)

**Bounded-error polynomial time (BPP)** to klasa problemów decyzyjnych, które można rozwiązać w czasie wielomianowym, z prawdopodobieństwem pomyłki co najwyżej  $\frac{1}{3}$  dla dowolnej z instancji.

### Definicja. (BQP)

**Bounded-error quantum polynomial time (BQP)** to klasa problemów decyzyjnych, które można rozwiązać w czasie wielomianowym na komputerze kwantowym, z prawdopodobieństwem pomyłki co najwyżej  $\frac{1}{3}$  dla dowolnej z instancji.

Zachodzi następujące:

$$P \subseteq BPP \subseteq BQP \subseteq PSPACE$$

Na dzień dzisiejszy dla żadnego z powyższych zawierań nie wiadomo czy są one ostre czy słabe.

### Definicja. (Uniwersalny zbiór bramek)

Zbiór bramek  $S$  nazywamy **uniwersalnym** jeśli dla każdej macierzy unitarnej  $U \in \mathbb{C}^{N \times N}$  oraz  $\varepsilon > 0$  istnieje obwód  $C$  złożony wyłącznie z bramek z  $S$ , nieużywający dodatkowych kubitów, taki, że dla każdego stanu  $|\psi\rangle$  zachodzi

$$\|U|\psi\rangle - C|\psi\rangle\| \leq \varepsilon$$

Intuicyjnie zbiór bramek jest uniwersalny, gdy możemy zasymulować z dowolną precyzją dowolną operację  $U$  możliwą na komputerze kwantowym.

## 2.1 Amplitude Amplification

### 3 Kubityzacja

Macierze, których używamy w układach kwantowych muszą być unitarne. Co jednak w sytuacji, gdy chcemy wykonać operację niebędącą unitarną? Nie wszystko stracone - wystarczy zakodować wymarzoną przez nas macierz  $A$  w większej macierzy  $U_A$ , poprzez dorzucenie dodatkowych kubitów (tzw. *ancillas*):

$$U_A = \begin{pmatrix} A & * \\ * & * \end{pmatrix}$$

gdzie wartości  $*$  nas nie interesują, bo dotyczą *ancilla* kubitów. Nie interesują nas ich konkretne wartości, ale co ważne są one dobrane tak, że macierz całościowo jest unitarna.

#### Definicja. (Block-encoding)

Mówimy, że  $U_A$  to  $(\alpha, a, \varepsilon)$ -**block encoding** macierzy  $A$ , jeśli:

$$\|A - \alpha(|0\rangle_a \otimes \mathbb{I}_s) U_A (|0\rangle_a \otimes \mathbb{I}_s)\| \leq \varepsilon$$

gdzie  $\alpha \in \mathbb{R}^+$ ,  $\varepsilon \in \mathbb{R}^+$ ,  $|0\rangle_a$  oznacza  $|0\rangle^{\otimes a}$

Zapis  $(|0\rangle_a \otimes \mathbb{I}_s) U_A (|0\rangle_a \otimes \mathbb{I}_s)$  oznacza tyle, że "wycinamy" z  $U_A$  fragment reprezentujący macierz  $A$ . Powyższa definicja jest dość ogólna. Zezwalamy by nasza macierz różniła się od wynikowej (w normie spektralnej) o co najwyżej  $\varepsilon$ . W definicji uwzględniamy również pewne przeskalowanie macierzy  $A$  o współczynnik  $\alpha$ . Do samego zakodowania macierzy  $U_A$  potrzebujemy też  $a$  *ancilla*-kubitów.

**Przykład.** Dla macierzy unitarnej  $A$ , samo  $A$  jest jej  $(1, 0, 0)$ -block encoding'iem

#### Lemat

Niech  $U_A$  i  $U_B$  to odpowiednio  $(\alpha, a, \varepsilon_A)$ -BE dla macierzy  $A$  i  $(\beta, b, \varepsilon_B)$ -BE dla macierzy  $B$ .

Niech  $U_{AB} := (I_b \otimes U_A)(I_a \otimes U_b)$ .

Wtedy  $U_{AB}$  jest  $(\alpha\beta, a+b, \alpha\varepsilon_B + \beta\varepsilon_A + \varepsilon_A\varepsilon_B)$ -BE dla macierzy  $AB$ .

### 3.1 LCU

**Linear Decomposition of Unitaries** (LCU) to metoda wyznaczania block-encoding'u dla zadanej macierzy  $A$ . Sam algorytm jest dość nieskomplikowany. Najpierw, zapiszmy  $A$  jako sumę macierzy unitarnych  $U_k$ :

$$A = \sum_{k=0}^{N-1} \alpha_k U_k$$

gdzie  $N = 2^{n+a}$ ,  $\alpha_i \in \mathbb{R}$ .

Teraz zdefiniujmy operatory *PREP* i *SEL*:

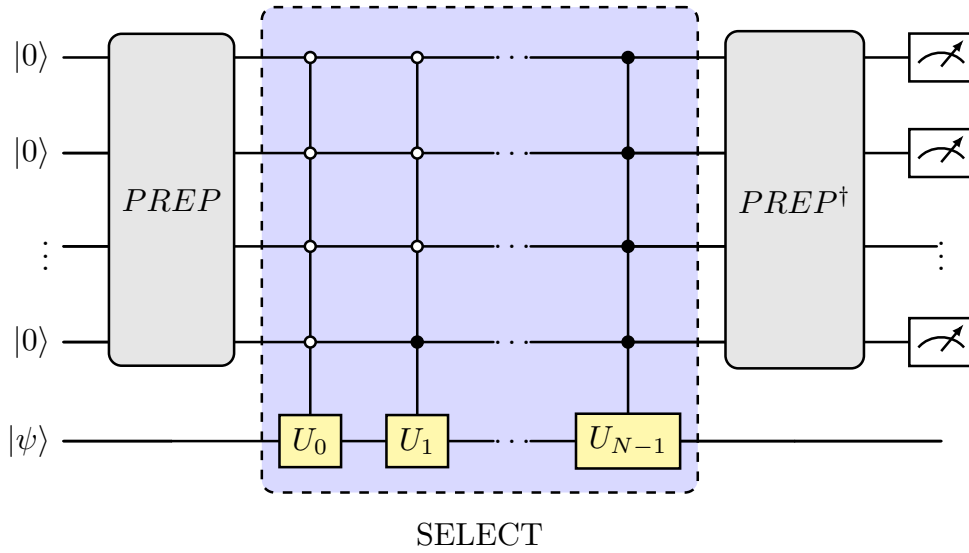
$$\begin{aligned} PREP |0\rangle &= \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} |k\rangle \\ SEL |k\rangle |\psi\rangle &= |k\rangle U_k |\psi\rangle \end{aligned}$$

gdzie współczynnik normalizacji  $\lambda$  jest zdefiniowany jako  $\lambda = \sum_{k=0}^{N-1} |\alpha_k|$ .

Zdefiniujmy teraz  $U_A$  jako

$$U_A := PREP^\dagger \cdot SEL \cdot PREP$$

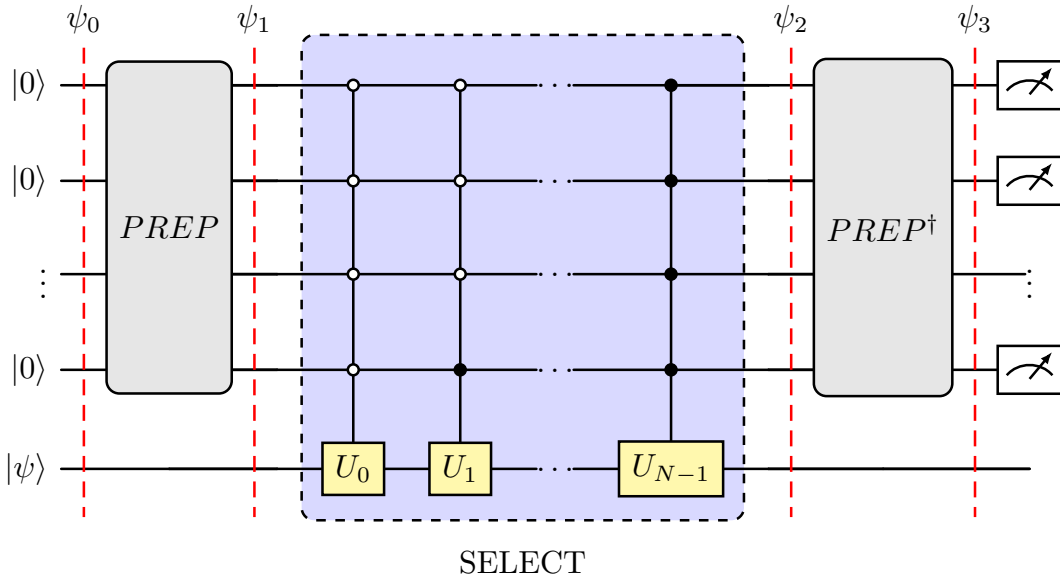
Okazuje się, że  $U_A$  jest  $(\lambda, a, 0)$ -block encoding'iem  $A$ . Oznacza to, że możemy skonstruować taki układ:



(6)

Jeśli teraz go uruchomimy, oraz na wszystkich (górnych)  $a$  miernikach zmierzmy  $|0\rangle$ , to końcowy stan na ostatniej (dolnej) linii będzie równy  $\frac{A}{\lambda} |\psi\rangle$ .

**Dowód.** Przeanalizujmy dokładnie co się dzieje w tym układzie



(7)

- $|\psi_0\rangle = |0\rangle_a |\psi\rangle$
- Aplikujemy  $PREP$ :

$$|\psi_1\rangle = PREP |0\rangle |\psi_0\rangle = \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} |k\rangle |\psi\rangle$$

- Aplikujemy  $SEL$ :

$$|\psi_2\rangle = SEL |\psi_1\rangle = \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} SEL |k\rangle |\psi\rangle = \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} |k\rangle U_k |\psi\rangle$$

- Aby poznać  $|\psi_3\rangle$  przyjrzyjmy się jak działa  $PREP^\dagger$ :

$$\langle 0 | PREP^\dagger = (PREP |0\rangle)^\dagger = \left( \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} |k\rangle \right)^\dagger = \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} \langle k |$$

Uzbrojeni w tą wiedzę dostajemy, że:

$$|\psi_3\rangle = \langle 0 | PREP^\dagger \cdot |\psi_2\rangle = \left( \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} \langle k | \right) \cdot \left( \sum_{k=0}^{N-1} \sqrt{\frac{|\alpha_k|}{\lambda}} |k\rangle U_k |\psi\rangle \right)$$



Pogrupujmy pary wartości z obu sum względem tego czy są te same:

$$= \left( \sum_i \frac{|\alpha_i|}{\lambda} \langle i|i \rangle U_i |\psi \rangle \right) + \left( \sum_{i \neq j} \frac{\sqrt{|\alpha_i| \cdot |\alpha_j|}}{\lambda} \langle i|j \rangle U_j |\psi \rangle \right)$$

Zauważmy, że skoro  $|i \rangle \perp |j \rangle$  to  $\langle i|i \rangle = 1$  oraz  $\langle i|j \rangle = 0$  dla  $i \neq j$ . Kasuje się prawy nawias i otrzymujemy:

$$= \sum_{i=0}^{N-1} \frac{|\alpha_i|}{\lambda} U_i |\psi \rangle = \frac{1}{\lambda} \sum_{i=0}^{N-1} \alpha_i U_i |\psi \rangle = \frac{A}{\lambda} |\psi \rangle$$

□

## 4 QFT

**Kwantowa transformata Fouriera** (ang. Quantum Fourier Transform) to przekształcenie analogiczne do Dyskretnej Transformaty Fouriera (DFT), tylko w przypadku kwantowym. Przypomnijmy najpierw definicję DFT:

### Definicja. (DFT)

Dyskretną transformatą Fouriera nazywamy funkcję

$$DFT : (x_0, \dots, x_{N-1}) \mapsto (y_0, \dots, y_{N-1})$$

gdzie

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega^{-jk}$$

$$\omega = \sqrt[N]{1} = e^{2\pi i/N}$$

Podobnie możemy sformułować QFT.

### Definicja. (QFT)

Niech  $|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ .

Kwantową transformatą Fouriera nazywamy funkcję

$$QFT : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{xk} |k\rangle$$

Można jednak patrzeć na to w inny sposób. Niech macierz Fouriera  $F_N$  będzie określona następująco:

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

Wtedy, przyjmując  $X = (x_0, \dots, x_{N-1})$ ,  $Y = (y_0, \dots, y_{N-1})$  mamy, że:

$$DFT(X) = X \cdot F_N = Y$$

Łatwo można sprawdzić, że macierz  $F_N$  jest unitarna. Zatem jest ona poprawną bramką kwantową. Oznacza to ni mniej ni więcej tyle, że QFT można traktować właśnie jako zaaplikowanie bramki  $F_N$ :

$$QFT |x\rangle \iff F_N |x\rangle$$

Teraz postaramy się skonstruować obwód dla  $F_N$ . Niech zapis  $0.k_1k_2\dots k_n$  oznacza liczbę zapisaną binarnie, przy pomocy bitów  $k_1, \dots, k_n$ . Wtedy zachodzi:

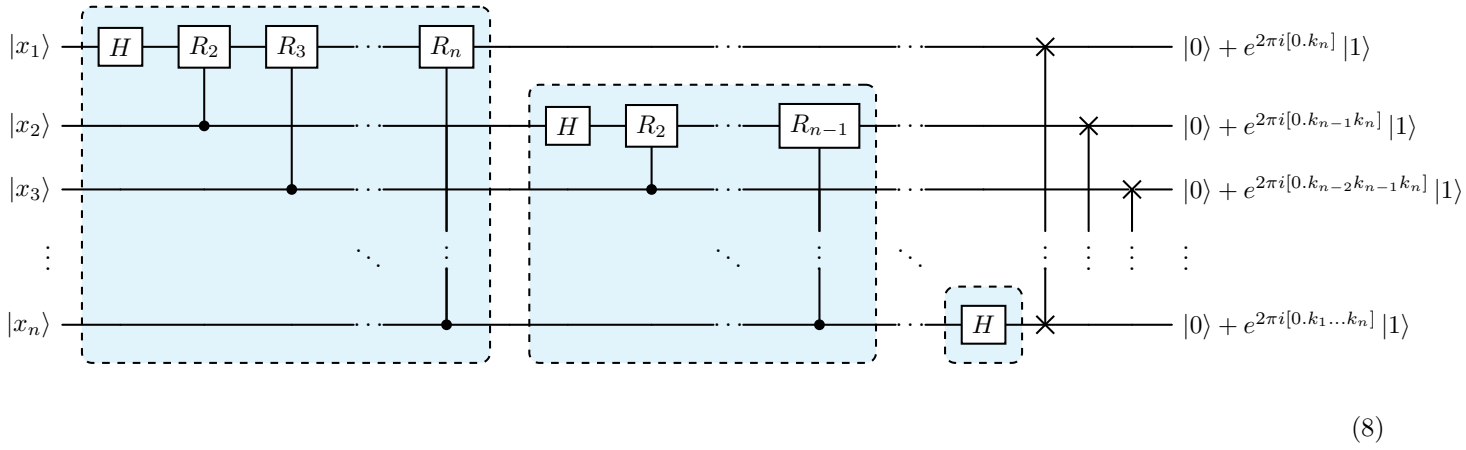
$$\begin{aligned} QFT |k\rangle &= QFT |k_1\rangle \dots |k_n\rangle \\ &= \frac{1}{\sqrt{N}} (|0\rangle + e^{2\pi i \cdot 0.k_n} |1\rangle) \oplus (|0\rangle + e^{2\pi i \cdot 0.k_{n-1}k_n} |1\rangle) \oplus \dots \oplus (|0\rangle + e^{2\pi i \cdot 0.k_1\dots k_n} |1\rangle) \end{aligned}$$

Reprezentacja ta jest o tyle pomocna, że przedstawiamy wyjście jako produkt tensorowy pojedynczych kubitów. Będą nam teraz potrzebne jedynie dwie bramki:  $H$  i  $R_k$ :

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$R_k = \begin{pmatrix} 1 & 0 \\ 1 & e^{2\pi i/2^k} \end{pmatrix}$$

Wtedy nasz układ prezentuje się następująco (na wyjściach dla czytelności pominięty współczynnik  $\frac{1}{\sqrt{2}}$ ):



**Dowód.** Zauważmy, że przy naszych oznaczeniach  $H = \frac{1}{\sqrt{2}} R_1$ . Zatem, jeśli prześledzimy pojedynczy kabel dojdziemy do wniosku, że domnażając wszystko do siebie otrzymamy *prawie* to czego potrzebujemy. Prawie, bo wszystko będzie w odwrotnej kolejności. Dlatego potrzebujemy jeszcze na końcu zamienić bity miejscami bramkami *SWAP*.  $\square$

Nasz układ posiada  $n$  bramek  $H$ ,  $(n-1) + \dots + 2 + 1 = \frac{n(n-1)}{2} R_k$  oraz  $\lfloor \frac{n}{2} \rfloor$  *SWAP*. Sumarycznie mamy zatem  $\mathcal{O}(n^2)$  bramek.

Jest to znakomity wynik, gdyż w klasyczne DFT wymaga  $\mathcal{O}(n2^n)$  operacji.

## 4.1 Quantum Phase Estimation

### Problem

Dany jest unitarny operator  $U$  oraz stan własny  $|\psi\rangle$ . A zatem możemy go zapisać jako:

$$U |\psi\rangle = e^{2\pi i \phi} |\psi\rangle$$

Zadaniem jest wyznaczenie wartości  $\phi$ .

Zauważmy, że skoro  $\phi \in [0, 1]$  to możemy traktować  $|\phi\rangle$  jako stan odpowiadający zapisowi binarnemu  $\phi$ . Zobaczmy co się stanie po zaaplikowaniu QFT na nim:

$$QFT |\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} |k\rangle$$

Wygląda podobnie do naszego  $U$ , ale nie do końca. Zapamiętajmy jednak, że jak zobaczymy wyrażenie po prawej, to operacją  $QFT^\dagger$  możemy odzyskać  $\phi$ .

Mamy zatem superpozycję stanów  $U^k |\phi\rangle$ :

$$U^k |\psi\rangle = e^{2\pi i \phi k} |\psi\rangle$$

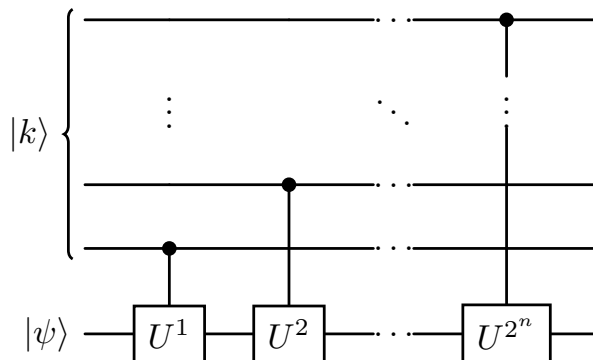
Zauważmy, że gdybyśmy znaleźli operator  $SEL'$  zachowujący się następująco:

$$SEL' |k\rangle |\psi\rangle = |k\rangle U^k |\psi\rangle$$

To umielibyśmy skonstruować cały układ:

$$|0\rangle_n |\psi\rangle \xrightarrow{H^{\otimes n} \otimes I} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |\psi\rangle \xrightarrow{SEL'} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle U^k |\psi\rangle = \frac{1}{\sqrt{2^n}} \left( \sum_{k=0}^{2^n-1} e^{2\pi i \phi k} |k\rangle \right) |\psi\rangle \xrightarrow{QFT^\dagger \otimes I} |\phi\rangle |\psi\rangle$$

Okazuje się, że  $SEL'$  nie jest tak trudny do skonstruowania:

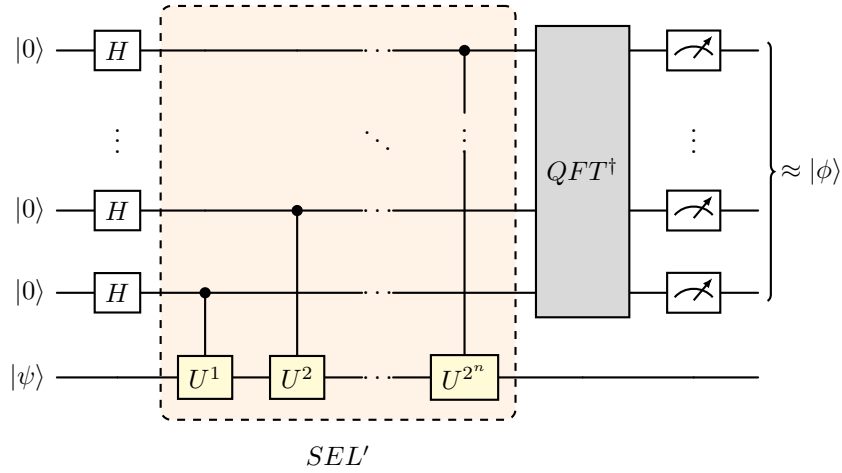


(9)

Jeśli przyjrzymy się ostatniemu kablowi i temu co do niego domnażamy to dojdziemy do wniosku, że faktycznie układ spełnia swoje zadanie. Stan  $|k\rangle$  odpowiada zapisowi binarnemu  $k$ , a zatem jeśli odpowiedni bit  $i$  będzie zapalony w  $k$  to

domnożymy do wyniku odpowiednią potęgę  $U^{2^i}$ .

Mamy już wszystko. Możemy zatem podziwiać końcowy układ w pełnej okazałości:



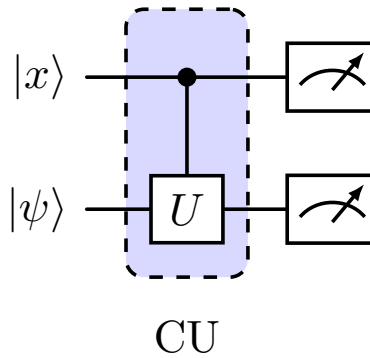
(10)

## 4.2 Phase kickback

### Definicja

**Phase kickback** to sytuacja, w której kubity kontrolowane mają wpływ na kubity je kontrolujące.

**Przykład.** Rozważmy dowolną bramkę CU (controlled unitary):



(11)

Jest kontrolowana, czyli będzie się zachowywać następująco:

$$CU |0\rangle |\psi\rangle = |0\rangle |\psi\rangle$$

$$CU |1\rangle |\psi\rangle = |1\rangle U |\psi\rangle$$

Niech dodatkowo  $|\psi\rangle$  będzie stanem własnym (eigenstate)  $U$ , czyli:

$$U |\psi\rangle = e^{2\pi i \phi} |\psi\rangle$$

Dla pewnego  $\phi$ . Ale uwaga, bo teraz jeśli popatrzymy dowolny stan  $|x\rangle = \alpha |0\rangle + \beta |1\rangle$  to otrzymujemy:

$$\begin{aligned} CU |x\rangle |\psi\rangle &= CU [(\alpha |0\rangle + \beta |1\rangle) \otimes |\psi\rangle] = \alpha CU(|0\rangle |\psi\rangle) + \beta CU(|1\rangle |\psi\rangle) \\ &= \alpha |0\rangle |\psi\rangle + \beta e^{2\pi i \phi} |1\rangle |\psi\rangle = (\alpha |0\rangle + \beta e^{2\pi i \phi} |1\rangle) |\psi\rangle \end{aligned}$$

Jeśli przypatrzymy się wejściu i wyjściu dojdziemy do ciekawego wniosku - "wynikowy" stan  $|\psi\rangle$  ma wpływ na wartość pierwszego kubit.

Właśnie tego typu efekt rozumiemy przez *phase kickback*. Sam w sobie niczemu nie służy. Ma on za zadanie jedynie tłumaczyć dlaczego czasami w obwodach występują rzekomo "zbędne" elementy, które mimo to mają wpływ na działanie całego układu.

## 4.3 Hamiltonian simulations

Jak wszyscy dobrze pamiętamy równanie Schrödingera wygląda z grubsza tak:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle$$

gdzie  $|\phi_t\rangle = |\phi(t)\rangle$ . Niech  $|\psi_0\rangle$  oznacza stan początkowy.

Zakładając, że  $H(t) = H$  to rozwiązanie przedstawia się następująco:

$$|\psi_t\rangle = e^{-itH/\hbar} |\psi_0\rangle = U(t) |\psi_0\rangle$$

Czyli:

$$U(t) = e^{-iHt}$$

Chcielibyśmy wyznaczyć  $U(t)$ . Jednak problemem jest rozmiar  $H$  oraz fakt, że chcemy policzyć  $e^H \dots$ . Otóż  $H$  ma rozmiary  $2^n \times 2^n$  co już dla dziesiątek kubitów daje olbrzymie macierze, z których jeszcze trzeba policzyć eksponens. Z tego powodu wystarczy nam wyznaczenie  $U'(t) \approx U(t)$ . Aby pokonać problemy obliczeniowe skorzystamy z faktu, że:

$$e^{-iHt} = \lim_{L \rightarrow \infty} \left( e^{-i\frac{t}{L}H_1} e^{-i\frac{t}{L}H_2} \right)^L$$

gdzie  $H = H_1 + H_2$ . Będziemy próbowali tak ustalić  $H_1, H_2$ , by łatwo dało się z nich policzyć exp. Na nasze potrzeby  $L < \infty$ , więc będą występować błędy. Niech  $\Delta t = \frac{t}{L}$ . Wtedy zachodzą następujące zależności:

$$\begin{aligned} \|e^{-i\Delta t H} - e^{-i\Delta t H_1} e^{-i\Delta t H_2}\| &= \mathcal{O}(\Delta t^2) \\ \|e^{-itH} - e^{-itH_1} e^{-itH_2}\| &= \mathcal{O}\left(\frac{t^2}{L}\right) \end{aligned}$$

# 5 Algorytm Shor'a

## Problem Simona

### Problem

Dana jest czarna skrzynka w postaci funkcji  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ ,  $m \geq n$ . Mamy gwarancję, że istnieje wartość  $s$ , dla której:

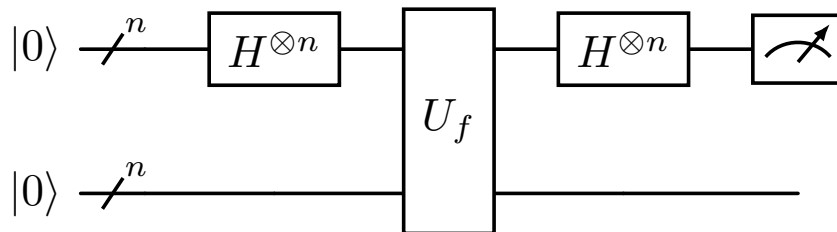
$$f(x) = f(x') \iff x = x' \oplus s$$

Zadaniem jest wyznaczenie wartości  $s$ .

Chcemy również wykonać możliwie mało ewaluacji funkcji  $f$ .

Klasyczne podejście polegałoby na ciągłym ewaluowaniu  $f$  i obserwowaniu, czy nie otrzymaliśmy uzyskanej już wcześniej wartości. Jednak mamy aż  $2^n$  argumentów do sprawdzenia. Nawet uwzględniając paradoks urodzin, rozwiązanie będzie wymagało średnio  $\mathcal{O}(\sqrt{2^n})$  ewaluacji funkcji  $f$ .

My jednak zaprezentujemy algorytm kwantowy, który pozwoli wyznaczyć  $s$  z dużym prawdopodobieństwem, przy użyciu  $\mathcal{O}(n)$  zapytań o funkcję  $f$ . Głównym elementem będzie następujący obwód:



(12)

Sam algorytm przedstawia się następująco:

1. Uruchom powyższy układ odpowiednią ilość razy, generując zbiór liniowo niezależnych wektorów bitowych  $y_1, \dots, y_{n-1}$ , **prostopadłych do  $s$** .
2. Każde wygenerowane  $y_k$  spełnia  $y_k \cdot s = 0$ . Z tego układu równań wyznaczamy wartość  $s$ .

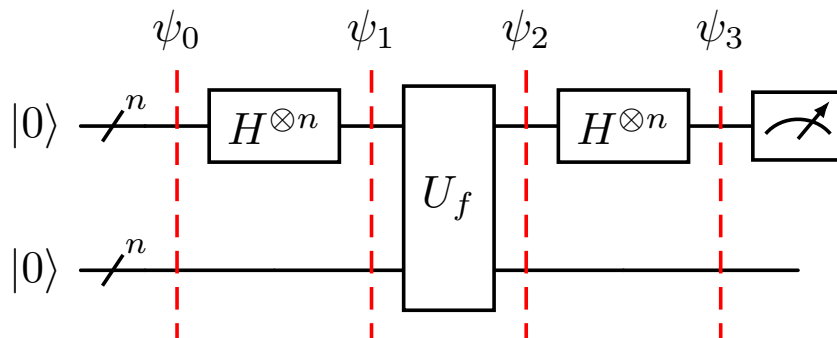
I to wszystko.

Pozostaje tylko kwestia, dlaczego to w ogóle działa. W tym celu pokażemy dwie rzeczy:

- (i) Każdy wektor wygenerowany przez układ jest prostopadły do  $s$ .
- (ii) Szansa, że wszystkie  $y_1, \dots, y_{n-1}$  są liniowo niezależne jest niemała.

### Dowód. (i)

W tym celu musimy przeanalizować działanie układu krok po kroku:



(13)

- $|\psi_0\rangle = |0\rangle_n |0\rangle_n$
- Bramka Hadamarda działa tak:

$$H |0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle$$

A więc:

$$|\psi_1\rangle = (H \otimes I_n) |\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |0\rangle_n$$

- Wyrocznia  $U_f$  działa tak:

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$$

Zatem:

$$|\psi_2\rangle = U_f |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle |f(k)\rangle$$

- Hadamard działa też tak:

$$H |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{k \cdot j} |j\rangle$$

To daje nam

$$\begin{aligned} |\psi_3\rangle &= (H \otimes I_n) |\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left[ \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{k \cdot j} |j\rangle \right] |f(k)\rangle \\ &= \sum_{j=0}^{2^n-1} |j\rangle \left[ \frac{1}{2^n} \sum_{k=0}^{2^n-1} (-1)^{j \cdot k} |f(k)\rangle \right] \end{aligned}$$

A zatem prawdopodobieństwo zmierzenia danego stanu  $|j\rangle$  jest równe:

$$\left\| \frac{1}{2^n} \sum_{k=0}^{2^n-1} (-1)^{j \cdot k} |f(k)\rangle \right\|^2$$

TODO... (na Wikipedii angielskiej jest reszta dowodu)

□

**Dowód. (ii)**

TODO... na Wikipedii angielskiej jest jakaś notka o tym sensowna

□

## 5.1 Algorytm Shora

### Problem

Dana jest liczba  $n$ .

Zadaniem jest znalezienie nietrywialnego jej dzielnika  $d$ .

Sam algorytm składa się z dwóch części:

1. Klasycznej, sprowadzającej problem faktoryzacji do problemu znajdowania rzędu<sup>1</sup> w grupie multiplikatywnej.
2. Kwantowej, znajdującej ten rząd w zadowalającym czasie

### Część klasyczna

W dalszej części będziemy zakładać, że  $N$  jest liczbą nieparzystą niebędącą potęgą liczby pierwszej.

**Obserwacja.** Aby sprawdzić czy  $N$  jest postaci  $p^k$  wystarczy przeiterować się po wszystkich sensownych  $k$ , tj.  $k \leq \lg N$  (dla  $k > \lg N$  mamy  $\sqrt[k]{N} < 2$ ). Dla każdego takiego  $k$  znajdujemy  $p' = \sqrt[k]{N}$  (np. wyszukiwaniem binarnym). Jeśli takowa liczba  $p'$  jest pierwsza (co sprawdzimy np. Millerem-Rabinem) to liczba  $N$  była postaci  $p^k$  dla  $p \in \mathbb{P}$ .

Wtedy algorytm Shora przedstawia się następująco:

1. Losujemy  $a \in [2, N-1]$
2. Wyliczamy  $\gcd(a, N)$ . Jeśli  $\gcd(a, N) > 1$  to zwróć  $d := \gcd(a, n)$
3. Wyznaczamy  $r$  takie, że  $a^r \equiv 1 \pmod N$   
Jeśli  $r$  jest nieparzyste, wróć do kroku 1.
4. Wyliczamy  $d := \gcd(N, a^{r/2} + 1)$   
Jeśli  $d$  jest trywialny, wróć do kroku 1.
5. Zwróć  $d$

Okazuje się, że dla  $N$  niebędącego potęgą liczby pierwszej

### Część kwantowa

Jedyną niewiadomą w algorytmie pozostaje kwestia jak szybko wyznaczać rząd danego elementu.

TODO

## 5.2 Test Hadamarda

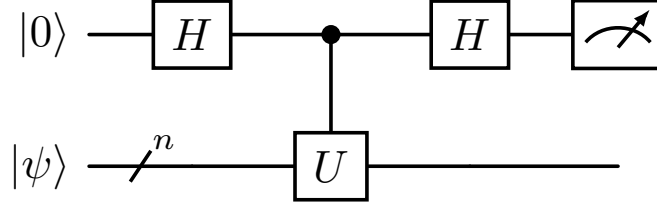
### Problem

Dany jest stan  $|\psi\rangle$  oraz operator  $U$ . Naszym celem jest wyznaczenie  $\text{Re} \langle \psi | U | \psi \rangle$ .

<sup>1</sup>Rzędem elementu  $a$  w grupie  $\text{mod } n$  nazywamy najmniejsze  $r > 0$  takie, że  $a^r \equiv 1 \pmod n$

Motywacją jest fakt, że często będziemy chcieli poznać jaką *dokładnie* liczbę zespoloną produkuje iloczyn skalarny  $\langle \phi | \psi \rangle$ . Można oczywiście bezpośrednio zmierzyć kubity na wyjściu, ale nie da nam to pełnej informacji, gdyż mierzymy jedynie amplitudę prawdopodobieństwa.

Test Hadamarda umożliwia nam stworzenie zmiennej losowej o wartości oczekiwanej równej  $\text{Re} \langle \psi | U | \psi \rangle$ . Można go zmodyfikować (dodając bramkę  $S^\dagger$ ) tak, by poznać też i wartość  $\text{Im}$ . Układ prezentuje się następująco:



(14)

Jeśli zmierzemy na wyjściu  $|0\rangle$  to wypływamy 1. Jeśli zmierzemy  $|1\rangle$  to wypływamy  $-1$ . Wtedy wartość oczekiwana na wyjściu jest równa dokładnie  $\text{Re} \langle \psi | U | \psi \rangle$ .

**Dowód.** Przeanalizujmy co się kolejno dzieje w naszym układzie.

Na wejściu mamy stan  $|0\rangle |\psi\rangle$ . Aplikujemy na górnym kablu Hadamarda, który działał tak:

$$H |0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle), \quad H |1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

Czyli u nas:

$$|0\rangle |\psi\rangle \xrightarrow{H \otimes I_n} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |\psi\rangle$$

Bramka  $U$  jest kontrolowana, czyli aplikuje się tylko dla  $|1\rangle$  na kontrolującym kubicie:

$$\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |\psi\rangle \xrightarrow{CU} \frac{1}{\sqrt{2}} (|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U |\psi\rangle)$$

Na końcu znowu aplikujemy Hadamarda:

$$\begin{aligned} \frac{1}{\sqrt{2}} (|0\rangle \otimes |\psi\rangle + |1\rangle \otimes U |\psi\rangle) &\xrightarrow{H \otimes I_n} \frac{1}{2} ((|0\rangle + |1\rangle) \otimes |\psi\rangle + (|0\rangle - |1\rangle) \otimes U |\psi\rangle) = \\ &= \frac{1}{2} (|0\rangle \otimes (|\psi\rangle + U |\psi\rangle) + |1\rangle \otimes (|\psi\rangle - U |\psi\rangle)) \\ &= \frac{1}{2} (|0\rangle \otimes (I + U) |\psi\rangle + |1\rangle \otimes (I - U) |\psi\rangle) \end{aligned}$$

Świetnie, zatem prawdopodobieństwo otrzymania  $|0\rangle$  to kwadrat ze współczynnika przy nim stojącego. W ogólności jeśli stoi tam wektor stanów  $|\phi\rangle$  to prawdopodobieństwo wynosi  $\langle \phi | \phi \rangle = |\phi\rangle^\dagger |\phi\rangle$ . Zatem liczymy prawdopodobieństwo  $p_0$  zmierzenia  $|0\rangle$ :

$$p_0 = \frac{1}{4} ((I + U) |\psi\rangle)^\dagger (I + U) |\psi\rangle = \frac{1}{4} \langle \psi | (I + U^\dagger) (I + U) |\psi\rangle$$

Jeśli powtórzymy analogiczne rozumowanie do policzenia prawdopodobieństwa  $p_1$  zmierzenia na wyjściu  $|1\rangle$  to otrzymamy, że:

$$p_1 = \frac{1}{4} \langle \psi | (I - U^\dagger) (I - U) |\psi\rangle$$

I teraz gwóźdź programu: traktujemy każde zmierzone  $|0\rangle$  jako 1, a każde  $|1\rangle$  jako  $-1$ . Oznaczmy tą zmienną losową przez  $X$ . Z definicji wartości oczekiwanej mamy:

$$\begin{aligned} E[X] &= p_0 \cdot (1) + p_1 \cdot (-1) \\ &= \frac{1}{4} \langle \psi | [(I + U^\dagger) (I + U) - (I - U^\dagger) (I - U)] |\psi\rangle \\ &= \frac{1}{4} \langle \psi | [\cancel{I} + U + U^\dagger + \cancel{U^\dagger U} - \cancel{I} + U + U^\dagger - \cancel{U^\dagger U}] |\psi\rangle \\ &= \frac{1}{2} \langle \psi | (U + U^\dagger) |\psi\rangle = \frac{1}{2} \langle \psi | U |\psi\rangle + \frac{1}{2} (\langle \psi | U |\psi\rangle)^\dagger \end{aligned}$$

Ale zauważmy, że  $\langle \psi | U |\psi\rangle = z \in \mathbb{C}$ . Zatem  $(\langle \psi | U |\psi\rangle)^\dagger = z^\dagger = \bar{z}$ . Ale z własności liczb zespolonych mamy, że  $z + \bar{z} = 2 \text{Re } z$ . Zatem:

$$E[X] = \frac{1}{2} (z + \bar{z}) = \text{Re } z$$

□

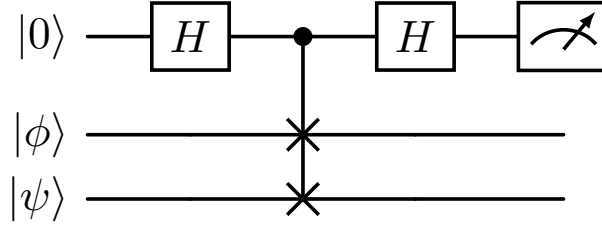


## 5.3 Test SWAP

### Problem

Dane są dwa stany  $|\phi\rangle$  oraz  $|\psi\rangle$ . Zadaniem naszym jest oszacowanie wartości  $|\langle\phi|\psi\rangle|^2$ .

Zauważmy, że test SWAP osiąga cel podobny do testu Hadamarda - pozwala nam ocenić, jak bardzo "podobne" są do siebie dwa stany  $|\phi\rangle$  i  $|\psi\rangle$ .  
Sam układ wygląda tak:



(15)

Główna bramka w nim występująca to kontrolowana bramka SWAP, która zamienia dwa dolne wejścia miejscami jeśli zostanie  $|1\rangle$  na górnym kablu.

Jeśli każde zmierzone  $|0\rangle$  potraktujemy jako 1, a każde  $|1\rangle$  jako  $-1$ , to okaże się, że wartość oczekiwana takiej zmiennej losowej wynosi  $|\langle\phi|\psi\rangle|^2$ .

**Dowód.** Prześledźmy kolejne aplikacje bramek w naszym układzie:

$$\begin{aligned} |0\phi\psi\rangle &\xrightarrow{H\otimes I\otimes I} \frac{1}{\sqrt{2}}(|0\phi\psi\rangle + |1\phi\psi\rangle) \xrightarrow{CSWAP} \frac{1}{\sqrt{2}}(|0\phi\psi\rangle + |1\psi\phi\rangle) \\ &\xrightarrow{X\otimes I\otimes I} \frac{1}{2}((|0\rangle + |1\rangle)|\phi\psi\rangle + (|0\rangle - |1\rangle)|\psi\phi\rangle) = \frac{1}{2}|0\rangle(|\phi\psi\rangle + |\psi\phi\rangle) + \frac{1}{2}|1\rangle(|\phi\psi\rangle - |\psi\phi\rangle) \end{aligned}$$

Liczymy prawdopodobieństwo  $p_0$  uzyskania  $|0\rangle$ :

$$\begin{aligned} p_0 &= \frac{1}{4}(|\phi\psi\rangle + |\psi\phi\rangle)^\dagger (|\phi\psi\rangle + |\psi\phi\rangle) = \frac{1}{4}(\langle\psi\phi| + \langle\phi\psi|)(|\phi\psi\rangle + |\psi\phi\rangle) \\ &= \frac{1}{4}(\langle\psi\phi|\phi\psi\rangle + \langle\psi\phi|\psi\phi\rangle + \langle\phi\psi|\phi\psi\rangle + \langle\phi\psi|\psi\phi\rangle) \\ &= \frac{1}{4}(1 + |\langle\phi|\psi\rangle|^2 + |\langle\phi|\psi\rangle|^2 + 1) \\ &= \frac{1}{2} + \frac{1}{2}|\langle\phi|\psi\rangle|^2 \end{aligned}$$

gdzie  $\langle\psi\phi|\psi\phi\rangle = \langle\phi\psi|\phi\psi\rangle = |\langle\phi|\psi\rangle|^2$  bierze się stąd:

$$\langle\psi\phi|\psi\phi\rangle = \langle\psi|\langle\phi|\psi\rangle|\phi\rangle = \langle\psi|\phi\rangle\langle\phi|\psi\rangle = (\langle\phi|\psi\rangle)^\dagger\langle\phi|\psi\rangle = |\langle\phi|\psi\rangle|^2$$

Z tego mamy, że prawdopodobieństwa  $p_1$  otrzymania  $|1\rangle$ :

$$p_1 = \frac{1}{2} - \frac{1}{2}|\langle\phi|\psi\rangle|^2$$

Wprowadźmy teraz naszą zmienną losową  $X$ . Z prawdopodobieństwem  $p_0$  otrzymujemy 1, a z prawd.  $p_1$  dostajemy  $-1$ . Pozostaje wyznaczyć  $E[X]$ :

$$E[X] = p_0 \cdot (1) + p_1 \cdot (-1) = \frac{1}{2} + \frac{1}{2}|\langle\phi|\psi\rangle|^2 - \frac{1}{2} + \frac{1}{2}|\langle\phi|\psi\rangle|^2 = |\langle\phi|\psi\rangle|^2$$

□

### Lemat

Dla ustalonej precyzji  $\varepsilon$  zarówno test Hadamarda jak i test SWAP wymagają  $N = \mathcal{O}(1/\varepsilon^2)$  prób.

**Dowód.** Bierze się to z faktu, że oba  $N$  razy próbują pewną zmienną losową, która zwraca jeden z możliwych wyników ze stałym prawdopodobieństwem  $p$ . Dla pojedynczej próby  $X_i$  wariancja  $\text{Var}[X_i]$  jest stała, tj.  $\text{Var}[X_i] = \mathcal{O}(1)$ . Każda próba jest niezależna od pozostałych, czyli dla  $X = \frac{1}{N}(X_1 + \dots + X_N)$  mamy:

$$\text{Var}[X] = \text{Var}\left[\frac{1}{N}(X_1 + \dots + X_N)\right] = \frac{1}{N^2}\text{Var}[X_1 + \dots + X_N] = \frac{1}{N^2}\sum_{i=1}^N \text{Var}[X_i] = \frac{1}{N^2}\mathcal{O}(N) = \mathcal{O}\left(\frac{1}{N}\right)$$

A odchylenie standardowe  $\sigma(X) = \sqrt{\text{Var}[X]} = \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ . Czyli chcąc  $\sigma(X) \leq \varepsilon$  potrzebujemy:

$$\sigma(X) \leq \varepsilon \iff \mathcal{O}\left(\frac{1}{\sqrt{N}}\right) \leq \varepsilon \iff \mathcal{O}\left(\frac{1}{N}\right) \leq \varepsilon^2 \iff N = \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$$

## 5.4 Amplitude estimation

Okazuje się, że można lepiej niż  $\mathcal{O}(1/\text{varepsilon})$ .

## 6 Stabilizatory

### Definicja

Wyróżniamy cztery **bramki Pauliego**:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

### Definicja

**Grupą Pauliego** dla pojedynczego kubitów nazywamy grupę

$$P_1 = \{\pm 1, \pm i\} \times \{X, Y, Z, I\}$$

Definicja działa również dla większej ilości kubitów:

$$P_n = \{\pm 1, \pm i\} \times \{X, Y, Z, I\}^{\otimes n}$$

### Definicja

Mówimy, że operator unitarny  $U$  **stabilizuje** stan  $|\psi\rangle$ , jeśli  $U|\psi\rangle = |\psi\rangle$ .

Wtedy  $U$  nazywamy **stabilizatorem**.

Równoważnie można powiedzieć, że stan  $|\psi\rangle$  jest **stabilizowany** przez operator  $U$ .

### Definicja

Niech  $|\psi\rangle$  to dowolny (określony) stan  $n$ -kubitowy.

Wtedy **grupa stabilizująca**  $G$  to podzbiór wszystkich  $P_n$  stabilizujących  $|\psi\rangle$ .

### Definicja

Dla  $n$ -kubitów, mianem **grupy stabilizującej Pauliego** określamy grupę  $G \leq P_n$ , która:

- jest abelowa ( $PQ \in G \iff QP \in G$ )
- $-I^{\otimes n} \notin G$

### Definicja

**Wymiar** grupy  $G$  definiujemy następująco:

$$\dim(G) := \log_2 |G|$$

W szczególności  $\dim(G) = k \in \mathbb{N}$  jest równoważne temu, że  $|G| = 2^k$ .

Ale przecież jeśli  $G \leq P_n$ , to z tw. Lagrange'a mamy, że  $|G|$  dzieli  $|P_n|$ . Ale (z definicji)  $P_n = 4^{n+1}$ . Oznacza to, że  $G = 2^k$ , dla pewnego  $k \in \mathbb{N}$ .

### Definicja

**Bramką Clifforda** nazywamy bramkę kwantową  $U$ , dla której zachodzi:

$$UPU^{-1} \in P_n$$

dla każdego  $P \in P_n$ .

**Fakt. (1)**  $Cliff_n = \langle H, S, CX \rangle$ .

Gdzie zapis  $\langle \dots \rangle$  oznacza grupę/zbiór generowane przez zadane bramki.

**Fakt. (2)** Zbiór bramek  $Cliff + T$  jest *uniwersalny*

### Twierdzenie. (Gotessman-Knill)

Obwody kwantowe złożone tylko z bramek Clifforda da się wielomianowo zasymulować na klasycznym komputerze.

## **7 Odporność na błędy**

## 8 Pamięć kwantowa